Database Engines for Geographical Information Systems

AGI

The Yearbook of the Association for Geographic Information

GEOGRAPHIC INFORMATION

1992/3

Edited by J. Cadoux-Hudson and D. I. Heywood

62

Database engines for Geographical Information Systems

Guy Haworth
ICL

Flashback

Melbourne, 9 April 1986, Simon Clarke reporting...

> The State Insurance Office (SIO) and police believe there may be a large-scale third-party insurance racket in Melbourne. Their investigations follow the discovery of more than 240 claims for whiplash injuries from addresses in an outer suburb area with a radius of one kilometre. Some 22 claims come from one small street and involve more than 40 per cent of the households there.
>
> The SIO had taken a 30 per cent increase in payments for some years and recorded a $671M loss for '84/5 on compulsory third-party insurance before it decided to take a closer look at the data. Now it believes fraudulent claims account for a "very significant proportion" of the losses. The 240 claims above could cost $5–12M on their own.

The SIO's core data-processing systems are not GIS systems in the accepted sense but the geographical element of their information was key to their discovery of a well-organized fraud. In fact, the SIO were only able to take a sideways look at their data and ask the right questions because they had just attached a "CAFS" file-searching engine to their central system.

How much money would they have saved if they had been able to identify the highly non-random geographical distribution of claimants years before? How much more would they have saved in staff time and payments if their customers had known they could analyse the data in any way they liked?

GIS on a grand scale

In January 1983, NASA launched IRAS, the "Infra-red Astronomical Satellite" into polar orbit 560 miles above the Earth. Within weeks, it became clear that the IRAS mission was the most comprehensive look at the universe ever undertaken. In only 10 months, the satellite increased the number of known extra-terrestrial objects by 40 per cent and identified 250,000 distinct radiation sources in the 600 megabytes of data sent back to base.

Back on the ground, the sheer volume of this raw data caused very real problems for the experts of the US, Holland and Britain. Until this mass of information had been analysed and interpreted from many angles, the secrets of the universe which IRAS had observed would remain unknown to humankind.

In the USA, the scientists went to work with the full might of their Cray supercomputers, taking 20 minutes of dedicated computer time with every question. In London's Queen Mary College, Dr. David Walker (Walker, 1985) shared the university's ICL 2988. However, he was able to scan the data in less than two minutes with a CAFS search engine and not surprisingly, he raced ahead of the American team developing his ideas iteratively on-line at will.

Dr. Walker not only discovered 10 000 galaxies but found that they were clustered in a particular direction. Now we know that we are actually situated in the outer suburbs of the known universe—filing interesting insurance claims.

Both of these stories indicate that a database engine, a system or subsystem designed to support database activity can, like CAFS, give orders of magnitude better access to the data. It can turn data into information into knowledge, and point to discoveries that could not otherwise be made.

The requirement for database engines

Before we look at specific technologies for and examples of database engines, we look at the requirements which are emerging today for database support. Organizations using GIS today include international organizations, national and local governments, the utilities and enterprises in the transport, retail and marketing sectors. All these organizations depend increasingly on their information resource to:

- reduce operational costs,
- improve effectiveness from stock-holding to customer service,
- support business development in new markets,
- create lasting competitive advantage in a rapidly changing context

They develop ever more comprehensive and detailed information models of their respective worlds of interest. They have also recognized the escalating demand for information access, driven by the following factors:

- growing volumes of information from increasingly many sources such as scanners, remote sensors, GPS, multiple databases, etc.
- growing interest in access of unformatted information such as text, maps, photographs, structured plans, multimedia, etc.
- the increasing trend to work on-line with quality colour desktop presentation, good communications, power, etc.
- the increasing number, literacy and ambition of information workers—TP, management support, statistical analysis, inference, etc.

This growing demand for information management is being met by a variety of technologies to acquire, store, access, analyse and present information and, by international standards, to reduce the visible variety of these technologies and help us work together.

Many technologies contribute to information management: "IRDS" data

dictionaries, transaction management systems, distributed processing software, wide-area communications and so on. Here we focus on the database engine itself, the business of storing, accessing and analysing information.

Information storage

The volume of raw data stored on computers, judged by the sales of disks, has been growing at 25-35 per cent per annum. In the GIS field, special factors are driving this growth. We now find graphics and image alongside data and text; a page of 4000 characters needs 40 Kbytes stored as graphics and 400 Kbytes as an image. Also, GIS data collection has recently been revolutionized in many ways including digitizers, scanners and remote sensing satellites.

NASA will be going for the database record circa 2000 with EOS, their Earth Observing System: 2 terabytes a day for 15 years = 11 000 terabytes = 11 petabytes, with 10 000 scientists seeking access to the data. If you are collecting geographical data, it's likely that you want to know where you are. The recent advent of the USA's Global Positioning System, GPS, offers new orders of efficiency and accuracy to surveyors and navigators worldwide, enabling more effective data collection.

Wherever this GIS data comes from, it has been collected at enormous expense and is literally priceless. Clearly it needs to be stored with security levels which would be the envy of a bank; no organization today can expect to survive the loss of its databases.

Further, it needs to be retrieved in a suitably short time for on-line users, regardless of the data volumes or transaction rates involved or the completeness of the data input to specify the query. As we move away from the classic COBOL-like data record to the less structured and more complex objects of GIS, we find the user asking higher-value and more wide-ranging questions, questions which incidentally make indexing techniques less relevant and information access more difficult.

All the above points away from the casual collection of all too portable PC discs and toward the "database engine"—a large scalable facility managed professionally on behalf of the organization and supporting large, growing and essentially indivisible collections of data. It is becoming clear that the main and enduring role of the mainframe or corporate server is as a database engine.

Information access

On-line systems potentially give information workers immediate access to the data they need. Today, the wider availability of suitable packages and "object oriented" applications development tools makes it far more likely that we will be working on-line.

On-line transactions can be characterized in terms of frequency and complexity, as judged by the load they place on the computer system; every computer system has its frontier of performance, see figure 1. Classic TP has high rate and low complexity; GIS transaction are much more likely to have medium and high complexity.

Information professionals in the GIS field will have an iterative and investigative style of on-line working like Dr Walker at QMC. They will be familiar with the short and predictable response times of classic TP systems and may wonder why their systems cannot currently achieve the same with their more complex queries. Where

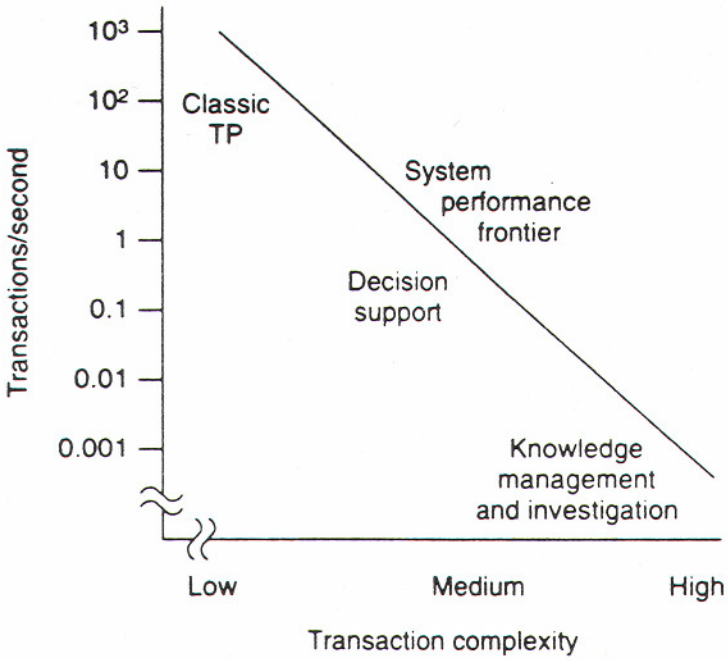

Figure 62.1. The range of transaction types.

"saga" transactions, like high-level query and statistical analysis, are the main workload, there will be a demand for response times closer to the classic two-second target.

Historically, as in the ICL, CAFS and NCR Teradata machines, the concept of parallelism has been employed to improve information access. Parallelism is relatively easy to use in the database field and we shall later look at the CEC Espirit II project EP2025, the European Declarative System, as a leading example of this approach.

Investment protection

Here, we examine how to protect your computer systems investment before and after exploiting database machines. We need to know how:

- best to prepare to use future database technology,
- to get best value from an investment in that technology.

The high cost and value of databases makes it vital that they are managed as long-life assets in an organization. Today, IT strategists are increasingly looking to build flexibility and survivability into their IT Systems by basing them on a framework of international standard interfaces. The "1992 effect" in Europe is just another step in the process of globalization which makes standards more important.

Standards allow existing systems to be insulated from and yet exploit new

technologies as they become available. Here, we look at the standards which create opportunities but impose constraints on the introduction of the next database engines.

The ISO SQL standard has emerged as the latest interface between application logic and database management. Most new conventional DP systems are now based on relational databases supporting SQL. SQL, SQL-86, was formally defined for the first time in 1986 and has been evolved through SQL-89 and SQL-92 to meet new requirements.

Even so, SQL-92 will remain firmly attached to its DP-data roots. We will have to wait for "SQL3" (SQL-x where x > 1995) before SQL formally manages complex GIS objects and knowledge with the higher level of intelligence which is also required (see Quinn, Abdelmoty and Williams, this volume). Steven Dowers has been representing AGI on ISO/IEC (ITC1 SC21 WG3) SQL Rapporteur Group and two AGI papers (Dowers, 1991; Gradwell, 1990) are a good indication of the GIS requirements influencing SQL3 extensions.

One of the big unknowns in the database future is whether SQL and industry-supplied DBMS technology can be evolved sufficiently to meet the needs of those, including the GIS community, dealing with non-classic complex data. The slow pace of ISO development and the wide variety of unsatisfied requirements makes it unlikely that SQL will be the only interface to GIS data.

However SQL will need to be supported in the GIS context. The established Relational DBMS vendors are extending their products beyond ISO SQL to meet the most immediate demands and GIS systems will need to cross-refer to information behind the SQL interface.

The new wave of object-oriented OODBMS technologies, primarily developed for CAD use, show an alternative approach which will certainly be adopted by the GIS community. The risk of using OODBMS technology will in the long term be reduced by standardization, most likely based on the work of OMG, the Object Management Group.

The reality therefore is that in future, a GIS will need to get its data from more than one type of DBMS and via more than one interface. It is also likely, given that different agencies collect data on different aspects of the same geography, that the databases will be owned by more than one information supplier and be physically distributed.

With this in mind, X/Open and ISO have devised the OSI-TP architecture to enable a transaction manager to interwork coherently with several database "resource" managers. These might be distributed over a number of computers, sites and organizations. The fact that the data you need is locked in a variety of technologies need not be a barrier to accessing that data.

It is possible to use Distributed TP software to advantage to implement TP systems involving one actual DBMS—and one hypothetical database manager. Those systems can then be enhanced more easily to exploit new database platforms as they become available.

What are the implications of these technologies and standards for the next database engines? To give flexibility, they are likely to be a mixture of software and hardware. It is important that the various items of software can be effectively ported to the engine's hardware. The engine should certainly support the SQL interface and be able to support both the established RDBMS packages and emerging database technologies. The database engine will be a "server", providing data to a variety of "client" applications, and will therefore support the necessary interfaces for distributed application processing.

Database engines today

Suppliers are providing a wider variety of computers today; there is clear recognition that general-purpose computers are going to lose market share to computers designed for a specific purpose. Signals from product trends, development projects and current research indicate that the database engine will be the target product for more than one company.

ICL's CAFS engine, already mentioned, and the NCR Teradata machine are the most conspicuous examples of commercial products today. The CEC Esprit II project EP2025 EDS, the European Declarative System, is in many ways a logical development of the thinking behind CAFS and has features in common with the Teradata, Meiko and Ncube machines. I shall therefore review CAFS and EDS as two generations of related database engine.

The CAFS engine

The purpose of CAFS (Haworth, 1985) is to filter and analyse data coming off a disk at disk speed. The requirement for such a search engine can be simply stated. Suppose you know that the answer to your question lies somewhere in a Gigabyte file of data. You could search it all but you would prefer if possible to avoid this since many disk-accesses slow down processing. You create one or more indexes to the data, chosen to help the queries you have in mind as much as possible. These indexes focus the following file search when relevant but they are costly to create, store and maintain and clearly have their limitations. For example:

- a full word-level inversion of a text file may be three to four times the size of the original text;
- would you really create an index of:

  — houses with at least seven of a given set of 10 features,
  — plant over five years old,
  — men whose eyes are not green,
  — addresses containing the letters "EAD" somewhere?

So we know that to access data, which we will do often, we always have to retrieve some part of the file and may have to search all of it, just to find a fragment of information.

The CAFS search engine was therefore implemented as an integral part of ICL's Series 39 Corporate Server architecture. It recognizes the data in the bit-stream, evaluates fields, selects records of information meeting a given criteria and returns only the interesting parts of those records.

The CAFS engine produces response-time improvements of 10–1000, the equivalent of 10–30 years development of conventional computer technology. Significantly in the context of GIS, the "turbo factor" is greater the more unstructured, and less indexable the information.

The CAFS concept has recently been re-implemented on ICL's Series 39 and DRS6000 UNIX machines. Searching is now done at up to 8 MB/sec and new applications include "CAFS" support for the INGRES RDBMS.

CAFS in practice

We have two CAFS stories already and there is only space for a few headlines and literature references on others (ICL, 1985; Walker, 1985; Wiles, 1985).

Southern Water (Corbin, 1985), leaders in giving their end-users large-scale access to their information, won the Office Automation award in 1985 for the "best information storage and retrieval system"; 80 per cent of their CAFS traffic was new work and included queries on rainfall, water quality and plant.

The Inland Revenue's National Tracing System (Wiles, 1985) routes mail to the correct tax office even when only a few characters of handwriting are decipherable. At peak times, 20 transactions/sec access a duplexed 12 GByte file, some 50 million name/address pairs. Two-thirds of the transactions require an index-focused CAFS scan which delivers a response in two to three seconds. NTS uses 26 CAFS engines in parallel to achieve this.

CAFS has been invaluable for the investigative systems needed by the intelligence, defence and police organizations. "Location" is one of the three most important items of information in such systems. During the UK privatization of the gas, water and electricity utilities, forensic accountants checked out share applications against the "one per person" rule, foiling a £6M raid on one occasion where a ring of over 40 people shared each other's addresses.

Social security fraud has similar name/address characteristics; back in Australia, one pilot, based on only seven GB of data, overcomes the "state boundary" problem. When the system matched aliases and addresses, it sent identical letters to each alias asking them to attend the same meeting to discuss their situation; false identities disappeared without further prompting—no delays, no legal fees.

From CAFS to the next generation

CAFS has been widely used since its inception; the technology, features and benefits behind its success are worth noting and looking for when considering a second generation database engine:

- functions "shipped" to the best place in the computer architecture;
- use of inexpensive components and a new technology mix;
- use of parallelism (see Lake's chapter in this volume);
- quantum improvement in performance over specified workload;
- exploitation via interfaces already valued by the market;
- fit with customer's existing investments—hardware, software, data;
- improved performance for existing systems;
- new opportunities to create high value information from raw data;
- product scalable to meet growth in demand for performance.

In "relational terms", CAFS performs RESTRICT and PROJECT, two of the basic three operation on data records. Now the industry is moving to support the SQL interface fully with purpose-built database engines.

In the US, Britton-Lee introduced such a product some years ago and were followed by Teradata, now part of NCR, with more success. Both engines did little for standard transaction processing but chose to support decision-support and data analysis.

The EDS project

In Europe, Bull, ICL, Siemens and their jointly owned European Computer Research Centre, ECRC, identified a common interest in 1988 in building a database machine and put a proposal to the CEC for Esprit II funding. The initial objectives (Haworth *et al.*, 1990) of the EDS project EP2025 will be met this year and further work will continue into 1993.

The messages from CAFS' success provided a good template of requirements for EDS which are to:

- improve performance more than 10-fold from TP to investigative query;
- support valued interface standards, e.g. XPG/POSIX base, C++, SQL;
- support established products and operate in client/server mode;
- maintain a near-linear relationship between size and power;
- adopt advances in hardware and software component technologies early;
- use parallelism to exploit the parallelism latent in data queries.

EDS exploits the fast-moving microprocessors and DRAM storage technologies on a large scale. Processors are increasing in nominal MIPS by 50 per cent per annum and DRAM chip storage is increasing in density by four every three to four years. At the end of 1992, the 80 MIPS processor will be in use and the 16 MBit chip will be adopted by the industry.

EDS aims for similar performance trends at the system level and the equivalent of several thousand TPCB transactions/second. In order to achieve this, EDS must be based on a distributed "share-nothing" architecture instead of the symmetric "share everything" architecture of current multiprocessor UNIX machines. In this arrangement of processors and stores, each processor "is near to" the part of the data in "its" store, the data layout is known to the DBMS, and the processors are asked to do the work related to their data.

To see how this works, imagine you ask a crowd of people if anyone has a birthday during the week. You have distributed your query to a "crowd machine"; each person has become a "processor" picking up part of the query, consulting their part of the "birthday database".

A key element determining the performance of a database machine like EDS is the communication system enabling the processors to work together. The Meiko Computing Surface uses a chess-board topology while the Teradata has its Y-net. The hardware and message-passing techniques of EDS have been developed only after extensive simulation by Siemens.

EDS will exploit parallelism in two ways. First, separate transactions can be executed simultaneously. All processing, even off-line batch processing, should therefore be thought of in transaction processing terms and divided up into a number of separate activities which can be run together. Second, a single transaction can be executed by a set of simultaneous processes. Parallelism within a transaction can be created by application logic or transparently by the optimizer within the RDBMS.

Just as ten CAFS engines can each look at 10 per cent of a file simultaneously, a hundred EDS processing elements could each look at 1 per cent of the data. In fact, any SQL query can be executed by a set of co-ordinated parallel activities as this query on plant-history and figure 2 illustrate:

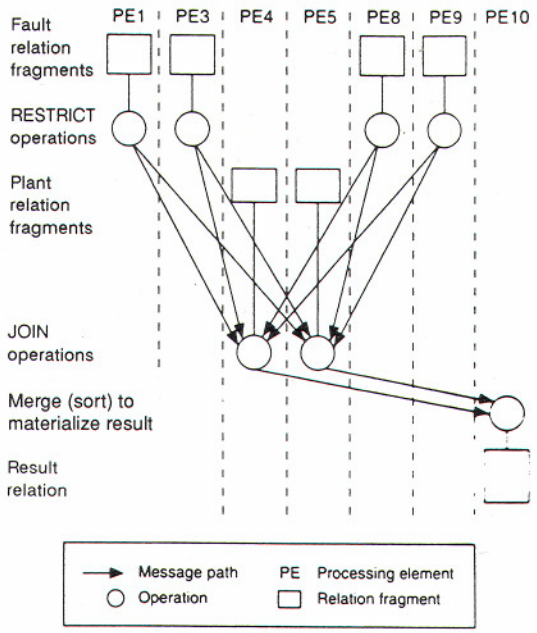

Figure 62.2. Parallel execution of a query.

Plant records (plant-id, location-id, plant-type ...) on processing elements 4 and 5

Fault records (plant-id, cause, ...) on elements 1, 3, 8 and 9

SELECT plant-id, location-id, cause FROM plant, fault
WHERE cause = electrical and plant-type = pump

By supporting the standardized POSIX and XPG operating system interfaces, EDS aims to be a platform for "UNIX" software and run RDBMS packages with much better performance.

In addition, the EDS research has developed database facilities for:

- support of user-defined data types and methods;
- support of complex objects and large objects;
- deductive database capabilities;
- general integrity constraints; and
- triggers (actions executed when a given event occurs).

With the above facilities, EDS will be able to manage complex information objects whose data content will not be simply retrieved but sometimes calculated or inferred from rules, e.g.:

- the age of a building;
- the fire-risk of a building, given a fire nearby;
- the safety envelope of an aircraft in flight;
- the predicted traffic load on a road, given a closure nearby.

The complementary concepts of object-orientation and deduction come together in the EDS database technology, enabling demanding GIS applications which require both to be supported naturally.

The EDS project demonstrates the potential of database engines based on a highly parallel architecture. You should be able to buy the database performance you are prepared to pay for; buy twice the hardware to double the throughput or halve the response times for complex queries.

The main partners in the EDS project are confident that the EDS approach to database has industrial potential. No products have yet been announced, but it is significant that all EDS-related projects proposed for CEC Esprit III funding, eight times oversubscribed, have been supported because of that potential.

Future scenarios

GIS applications require large-scale data management involving:

data acceptance:	integrity checking and calibration;
storage management:	distributing data to balance query load;
indexing:	characterizing the data by features useful for queries;
inference:	deriving higher-order information from raw data;
dissemination:	passing new data to interested parties.

Quinn *et al.* (in this volume) picks out map design, geographic database management and decision support, and automatic feature extraction as four key GIS applications. All of these are data and processing-intensive and most of the processing is within the database management part of the application. They will clearly benefit from the increased power of the database engines coming to market today. When these applications are required by mobile, real time or military situations with demanding availability and integrity requirements, the database engines required double and redoubles in required performance.

Perhaps this is a glimpse of the future

The Street Works Register

The government's Road and Street Works Act 1991 has resulted in the Street Works Register, a source of accurate definitions of roads, equipment above and below the surface, and repair history. The days of spatial data, hedged around with disclaimers on accuracy, are over.

The immediate beneficiaries have been local authorities and the utilities, particularly water companies who have to dig deepest to maintain their equipment. Their insurance claims for incidental damage to other unexpected equipment are radically down.

Significant savings on works costs amounting to many millions of pounds per year are being made and at the same time the country's street infrastructure is improved in terms of road quality and reduced delays to traffic and people. The database engines behind this advance provide a wide variety of information to all the agencies involved in street works and interested in definitive geographical data.

Heathrow, Friday 20.30

The computers of air-traffic control manage detailed data on each flight in the zone. Aircraft navigation systems radio in position data which is checked against past history and ground-radar data. The DBMS computes a safety zone for each plane. Flight paths on current instructions are simulated to check for near-miss situations.

The incoming shuttle from Manchester is diverted to an outlying station of terminal one. The DBMS infers that mobile steps and three buses are required. The nearest unassigned units arrive at the station just in time to meet the plane.

In the terminal, the displays show accurate forecasts of arrival and departure times, reflecting the fact that the DBMS behind them is fully informed of all delays and able to work out their impact.

Another "Gulf War"?

The last Gulf War was a sharp demonstration of the value of information technology and we can reasonably guess that the allied forces will have built on their success. Next time .....? Ground units communicate their GPS positions continuously, defining troop readiness and safety zones.

Overhead, remote sensors photograph the desert and send their results back to HQ. The DBMS picks out known features, registers the pictures, removes the images of friendly troops and, allowing for the time of day, checks the remainder for change against previous photographs. It infers that enemy troops are moving down a key road and that previously passable tracks of hard sand have been wiped out.

To avoid the dangers of information overload, the DBMS reports developments to the senior officers against the prioritized definition of their requirements.

Conclusions

Our ability to identify, acquire, store, enquire on and analyse data is increasing as never before, especially in the GIS field. Technologies are becoming available to manage a wider variety of data and to make intelligent inferences on that data.

The mainstream arrival of large-scale database engines is not far away. The experience of using the first such products tells us that they will radically change data management in the GIS field.

Abbreviations

C++	Object-oriented extension of the C language
CAD	Computer-Aided Design
CEC	Commission of the European Community
CAFS	Content-Addressable File Search (engine)
DBMS	DataBase Management System
DRAM	Dynamic Random Access (solid-state) Memory
DTP	Distributed TP
GB	Gigabyte, 10^9 bytes
GPS	(US.DoD) Global Positioning System
IRAS	Infra-Red Astronomical Satellite
IRDS	Information Resource Directory System (dictionary)
ISO	International Standards Organization
KB	KiloByte
MIP	One million instructions/second (of undefined type)
OODBMS	Object-Orientated DBMS
POSIX	Portable Operating System Interface for Computer Environments
Petabyte	10^{15} bytes
RDBMS	Relational DBMS
SQL	Structured Query Language
Terabyte	10^{12} bytes
TP	Transaction Processing
TPC	Transaction Processing Council, defines TP benchmarks
TPCB	The TPC's "B" benchmark for transaction servers
XA	The DTP interface between transaction and resource managers
X/Open	Worldwide body defining the Common Application Environment
XPG	The X/Open Portability Guide

Acknowledgements

My thanks to Chris Corbin, Steve Dowers, David Gradwell and Tom Lake for recent conversations and to my ICL colleagues for their contributions to CAFS and EDS.

References

Corbin, C.E.H., 1985, "Creating an end-user CAFS service", *ICL Technical Journal*, **4**, 4, (the "CAFS" issue) pp. 441-545.

Dowers, S., 1991, "SQL—The Way Forward", *AGI, Birmingham November 91*, pp. 3.13.1-5.

Gradwell, D.J.L., 1990, "Can SQL Handle Geographic Data?" A Presentation on the work of the AGI Standards Committee's SQL Working Party, AGI 1990 pp. D.3.1-12.

ICL Computer User Association, 1985 CAFS Special Interest Group, "CAFS in Action", November.

Haworth, G.McC., 1985, "The CAFS system today and tomorrow", *ICL Technical Journal*, **4**, 4, pp. 483-8.

Haworth, G.McC., Leunig, S., Hammer, C. and Reeve, M., 1990, "The European Declarative System, Database and Languages", *IEEE Micro*, Dec, pp. 20-23 and 85-88.

Walker, D., 1985, "Secrets of the sky: The IRAS data at Queen Mary College", *ICL Technical Journal*, **4**, 4, pp. 483-8.

Wiles, P.R., 1985, "Using secondary indexes for large CAFS databases", *ICL Technical Journal*, **4**, 4, pp. 419–40.